
Subject: [RFC][PATCH] Define/use task_pid_ns() wrapper
Posted by [Sukadev Bhattiprolu](#) on Sat, 10 Mar 2007 03:56:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [RFC][PATCH] Define/use task_pid_ns() wrapper

Use a wrapper to access/set pid_namespace of a task.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Cc: Cedric Le Goater <clg@fr.ibm.com>
Cc: Dave Hansen <haveblue@us.ibm.com>
Cc: Serge Hallyn <serue@us.ibm.com>
Cc: containers@lists.osdl.org

fs/exec.c | 2 +-
fs/proc/proc_misc.c | 2 +-
include/linux/pid_namespace.h | 11 ++++++++
kernel/exit.c | 4 +++-
kernel/pid.c | 6 +++-
5 files changed, 18 insertions(+), 7 deletions(-)

Index: lx26-20-mm2b/fs/exec.c

=====

--- lx26-20-mm2b.orig/fs/exec.c 2007-03-01 15:19:15.000000000 -0800
+++ lx26-20-mm2b/fs/exec.c 2007-03-01 15:20:52.000000000 -0800
@@ -620,7 +620,7 @@ static int de_thread(struct task_struct
 * so it is safe to do it under read_lock.
 */
 if (unlikely(tsk->group_leader == child_reaper(tsk)))
 - tsk->nsproxy->pid_ns->child_reaper = tsk;
 + task_pid_ns(tsk)->child_reaper = tsk;

zap_other_threads(tsk);
read_unlock(&tasklist_lock);
Index: lx26-20-mm2b/fs/proc/proc_misc.c

=====

--- lx26-20-mm2b.orig/fs/proc/proc_misc.c 2007-03-01 15:19:15.000000000 -0800
+++ lx26-20-mm2b/fs/proc/proc_misc.c 2007-03-01 15:20:52.000000000 -0800
@@ -93,7 +93,7 @@ static int loadavg_read_proc(char *page,
 LOAD_INT(a), LOAD_FRAC(a),
 LOAD_INT(b), LOAD_FRAC(b),
 LOAD_INT(c), LOAD_FRAC(c),
 - nr_running(), nr_threads, current->nsproxy->pid_ns->last_pid);
 + nr_running(), nr_threads, task_pid_ns(current)->last_pid);
 return proc_calc_metrics(page, start, off, count, eof, len);
 }

Index: lx26-20-mm2b/kernel/exit.c

```
=====
--- lx26-20-mm2b.orig/kernel/exit.c 2007-03-01 15:19:15.000000000 -0800
+++ lx26-20-mm2b/kernel/exit.c 2007-03-01 15:20:52.000000000 -0800
@@ -790,8 +790,8 @@ fastcall NORET_TYPE void do_exit(long co
     if (unlikely(!tsk->pid))
         panic("Attempted to kill the idle task!");
     if (unlikely(tsk == child_reaper(tsk))) {
- if (tsk->nsproxy->pid_ns != &init_pid_ns)
- tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
+ if (task_pid_ns(tsk) != &init_pid_ns)
+ task_pid_ns(tsk)->child_reaper = init_pid_ns.child_reaper;
     else
         panic("Attempted to kill init!");
     }
}
```

Index: lx26-20-mm2b/kernel/pid.c

```
=====
--- lx26-20-mm2b.orig/kernel/pid.c 2007-03-01 15:19:15.000000000 -0800
+++ lx26-20-mm2b/kernel/pid.c 2007-03-07 19:52:21.000000000 -0800
@@ -213,7 +213,7 @@ struct pid *alloc_pid(void)
     if (!pid)
         goto out;

- nr = alloc_pidmap(current->nsproxy->pid_ns);
+ nr = alloc_pidmap(task_pid_ns(current));
     if (nr < 0)
         goto out_free;

@@ -358,7 +358,7 @@ struct pid *find_ge_pid(int nr)
     pid = find_pid(nr);
     if (pid)
         break;
- nr = next_pidmap(current->nsproxy->pid_ns, nr);
+ nr = next_pidmap(task_pid_ns(current), nr);
     } while (nr > 0);

     return pid;
@@ -367,7 +367,7 @@ EXPORT_SYMBOL_GPL(find_get_pid);

int copy_pid_ns(int flags, struct task_struct *tsk)
{
- struct pid_namespace *old_ns = tsk->nsproxy->pid_ns;
+ struct pid_namespace *old_ns = task_pid_ns(tsk);
     int err = 0;

     if (!old_ns)
```

Index: lx26-20-mm2b/include/linux/pid_namespace.h

```

=====
--- lx26-20-mm2b.orig/include/linux/pid_namespace.h 2007-03-01 15:19:15.000000000 -0800
+++ lx26-20-mm2b/include/linux/pid_namespace.h 2007-03-07 19:52:21.000000000 -0800
@@ -37,6 +37,17 @@ static inline void put_pid_ns(struct pid
    kref_put(&ns->kref, free_pid_ns);
}

+static inline struct pid_namespace *task_pid_ns(struct task_struct *tsk)
+{
+    return tsk->nsproxy->pid_ns;
+}
+
+static inline void set_task_pid_ns(struct task_struct *tsk,
+    struct pid_namespace * ns)
+{
+    tsk->nsproxy->pid_ns = ns
+}
+
static inline struct task_struct *child_reaper(struct task_struct *tsk)
{
    return init_pid_ns.child_reaper;
}

```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
